BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-271792
(43)Date of publication of application : 20.10.1995

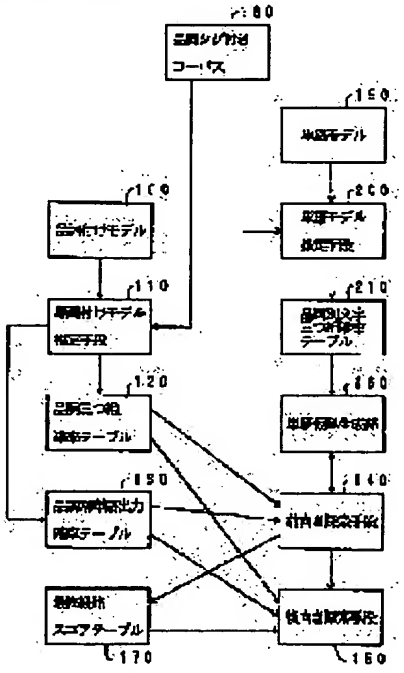
(51)Int.Cl. G06F 17/27

(21)Application number : 06-061527
(22)Date of filing : 30.03.1994

(71)Applicant : NIPPON TELEGR & TELEPH CORP <NTT>
(72)Inventor : NAGATA MASAAKI

(54) DEVICE AND METHOD FOR ANALYZING JAPANESE MORPHEME

(57)Abstract:
PURPOSE: To output only an optional number of pairs of word-divided and part-of-speech added candidates from the most likelihood candidate even when an input sentence includes an unknown word by degenerating a double Markokv process into a single Markov process in consideration of secondary expansion of a symbol, and handling part-of-speech triple probability in a frame obtained by expanding a conventional connection cost minimizing method.
CONSTITUTION: This device is provided with a part-of-speech model estimating means 110, a part-of-speech triple possibility table 120, a part-of-speech distinctive word output probability table 130, an minimum path score table 170, and a forward search means 140 and a backward search means 150. On the basis of a part-of-speech added model constituting of part-of-speech triple probability and part-of-speech distinctive word output probability, the forward search means 140 is used first to store the optimum path score table 170 with the maximum value of simultaneous probability of a word string and a part-of-speech string from the head of the sentence to one word, and on the basis of the value of the optimum path score table 170, the backward search means 150 is used to finds an optional number of morpheme analytic candidates in the most likelihood order.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

02RL148

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平7－271792

(43)公開日　平成７年(1995)10月20日

(51)Int.Cl.*	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　17/27				
		8219－5L	Ｇ０６Ｆ　15/ 38	Ｅ

審査請求　未請求　請求項の数４　ＯＬ　（全 18 頁）

(21)出願番号　特願平6－61527

(22)出願日　平成６年(1994)３月30日

(71)出願人　000004226
日本電信電話株式会社
東京都千代田区内幸町一丁目１番６号

(72)発明者　永田　昌明
東京都千代田区内幸町１丁目１番６号　日本電信電話株式会社内

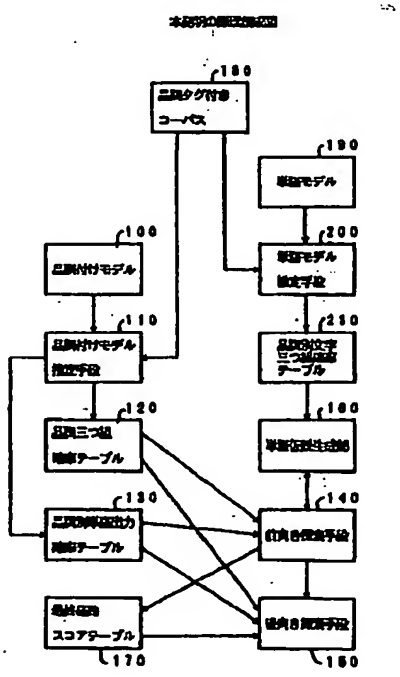
(74)代理人　弁理士　伊東　忠彦

(54)【発明の名称】　日本語形態素解析装置及び日本語形態素解析方法

(57)【要約】

【目的】　本発明の目的は、単語分割と品詞付与の組を、入力文が未知語を含む場合でも最も尤もらしい候補から順に任意の数だけ出力できるので、高い精度をもち、頑強でかつ柔軟なインタフェースを持つ日本語形態素解析装置及び日本語形態素解析方法を提供することである。

【構成】　本発明は、統計的な言語モデルである品詞付けモデルに基づいて、多重マルコフ過程の縮退、前向き動的計画法、内路向きA*探索を用いて、二回探索することにより、辞書に未登録の単語の形態素解析も含めて実現する。



【特許請求の範囲】

【請求項１】　与えれた日本語の入力文を形態素解析する装置であって、

連続する３つの品詞の組において直前の２つの品詞が与えられた時の３つ目の品詞の確率である品詞三つ組確率と、品詞が与えられた時の単語の確率である品詞別単語出力確率から、文を構成する単語列と各単語に付与された品詞列の同時確率を与える品詞付けモデルと、

単語に分割され、かつ、品詞が付与された文の集合から、品詞三つ組確率と品詞別単語出力確率を推定する品詞付けモデル推定手段と、

該品詞三つ組確率を記憶する品詞三つ組確率テーブルと、

該品詞別単語出力確率を記憶する品詞別単語出力確率テーブルと、

文頭からある単語に至るまでの単語列と品詞列の同時確率の最大値を記憶する最適経路スコアテーブルと、

該最適経路スコアテーブルに記憶されている該文頭からある単語に至るまでの単語列と品詞列の同時確率の最大値を、該単語を含む直前の２つの単語に付与された品詞の組が異なる場合毎に、該単語を含む直前の３つの単語に付与された品詞三つ組確率、該単語の品詞別単語出力確率、及び文頭からその単語の直前の単語に至るまでの単語列と品詞列の同時確率の最大値から求め、該最適経路スコアテーブルに記録する前向き探索手段と、

単語列と品詞列の同時確率を最大化するような入力文の単語分割及び品詞付与の候補を、文末からある単語に至るまでの単語列と品詞列の同時確率の最大値、及び該最適経路スコアテーブルに記録された、文頭からその単語に至るまでの単語列と品詞列の同時確率の最大値から、最も尤もらしい候補から順番に任意の個数だけ求める後向き探索手段とを有し、

該品詞三つ組確率と該品詞別単語出力確率から構成される該品詞付けモデルに基づいて、まず、該前向き探索手段を用いて、文頭からある単語までの単語列と品詞列の同時確率の最大値を該最適経路スコアテーブルに記憶し、該最適経路スコアテーブルの値に基づいて該後向き探索手段を用いて最も尤もらしい順番に任意の個数の形態素解析候補、即ち、単語列と品詞列の組を求めることを特徴とする日本語形態素解析装置。

【請求項２】　単語に分割され、かつ品詞が付与された文の集合から、品詞別文字三つ組確率を推定する単語モデル推定手段と、

該品詞別文字三つ組の確率から、該単語毎に品詞別出力確率を与える単語モデルと、

該品詞別文字三つ組確率を記憶する品詞別文字三つ組確率テーブルと、

ある文字位置から始まる入力文の部分文字列の中から、品詞別文字三つ組確率に基づいて、最も尤もらしい順番に任意の個数の単語仮説、即ち、表記、品詞及び品詞別出力確率の組を求める単語仮説生成部と、

該入力文中に未知語が存在する場合にも、前記前向き探索手段において、該品詞別文字三つ組から構成される該単語モデルを用いて、文字表記から単語仮説を生成し、前記後向き探索手段において、文全体を考慮して最も尤もらしい、未知語の単語区切り、品詞及び品詞別単語出力確率を求めることを含む請求項１記載の日本語形態素解析装置。

【請求項３】　日本語の文が入力されると、連続する三つの連続する品詞の組において直前の二つの品詞が与えられた時の三つ目の品詞の確率である品詞三つ組確率と品詞が与えられた時の単語の確率である品詞別単語出力確率に基づいて、文頭から文末へ一文字ずつ進む動的計画法を用いて、文頭からある単語に至るまでの単語列と品詞列の同時確率の最大値を求める前向き探索を行い、文末から文頭へ進むA*アルゴリズムを用いて、文を構成する単語列と各単語に付与された品詞列の同時確率を最大化する形態素解析候補、即ち、単語列と品詞列の組を最も尤もらしい順に一つずつ求める後向き探索を行う日本語形態素解析方法。

【請求項４】　入力文が辞書に登録されていない未知語を含む場合に、前記前向き探索において、前記品詞別文字三つ組確率に基づいて、単語を構成する文字列の始まりと終わり、品詞、品詞別出力確率からなる単語仮説を生成し、

前記後向き探索により最も尤もらしい単語分割と品詞付与を決定する請求項３記載の日本語形態素解析方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、日本語形態素解析装置及び日本語形態素解析方法に係り、特に、日本語文を単語に分割し、各単語に品詞を付与する日本語形態素解析装置及び日本語形態素解析方法に関する。

【０００２】詳しくは、品詞タグ付きコーパスから統計的な手法によって求めた言語モデルを用いることにより、文法や辞書を対象領域に自動的に適応化し、入力文の長さに比例する計算量で効率的に精度よく形態素解析を行い、任意の個数の最尤な形態素解析候補を求め、入力文が未知語を含む場合にも形態素解析が行えるようにしたものである。

【０００３】

【従来の技術】従来の日本語形態素解析システムは、文法規則と発見的探索に基づく方式が主流であり、これらは、文法規則（言語モデル）として品詞接続表を用いて、形態素解析候補の探索の際には、発見的規則（ヒューリスティックス(heuristics)）を用いて候補の順位付けを行う（第１の従来の方法）。

【０００４】形態素解析で用いられる発見的規則としては、入力文と照合する最も長い辞書中の単語を含む形態素解析候補を優先する最良一致法（第１の従来の方法）

や、文節数が最も少ない形態素解析候補を優先する文節数最小法（吉村・日高・吉田「文節数最小法を用いたべた書き日本語文の形態素解析」情処論 Vol. 24 No. 1, pp. 40-46, 1983)がある（第２の従来の方法）。

【０００５】また、文節数最小法を拡張し、解を詳細に順序付けする方法として、接続コスト最小法（久光・新田「接続コスト最小による形態素解析の提案と計算量の評価について」信学会 NLC90-8, pp. 17-24, 1990）がある。これは、接続コストが最小となる形態素解析候補を、動的計画法を用いて、入力文の長さに比例する計算量を求めることができるアルゴリズムである（第３の従来の方法）。

【０００６】また、近年、上記の従来の方法である「規則に基づく方法」に対して、「統計に基づく方法」が研究され始めている。これは、必ずしも根拠が明確ではなかった発見的な接続コストの代わりに、大量のテキストデータから求めた統計的言語モデルに基づく連接確率を使用する方法である（第４の従来の方法）。

【０００７】また、上記の第３の実施例の接続コスト最小法における接続コストの代わりに、確率正規文法に基づくコストを使用する方法が提案されている（松延・日高・吉田「確率文節文法による構文解析」情処 NL56-3, 1986）。この方法では、確率正規文法、即ち、品詞の連接確率と品詞別の単語の出力確率の情報を接続コストの計算に用いる。

【０００８】上記の統計的な言語モデルを用いることにより、形態素解析候補の優先度の根拠が明確により、かつ、候補の詳細な順序付けができるようになる。

【０００９】また、従来の日本語形態素解析システムにおいて辞書に登録されていない未知語の扱いは、非常に「場当たり的（ad hoc）」である。多くのシステムは、「同じ字種の文字の連続が単語を構成することが多い」、或いは、「カタカナの連続は外来語の名詞であることが多い」というような字種に関する発見的規則を用いて、単語の分割及び、品詞の付与を行っている（吉村・武内・津田・首藤「未登録語を含む日本語文の形態素解析」情処論 Vol. 30 No. 3, pp. 294-301, 1989)。或いは、付属語列等から文節を推定し、そこから、付属語を取り除いた部分列を未知語とみなす方法が使われていることが多い（第５の従来の方法）。

【００１０】未知語問題に対する統計的な解決策として、造語単位の２つの組を用いて単語の生起確率を推定する方法が提案されている（永井・日高「日本語における単語の造語モデルとその評価」情処論 Vol. 34 No. 9, pp. 1944-1955, 1993)。この方法は、単語辞書の見出し語から造語モデルのパラメータを推定する。

【００１１】

【発明が解決しようとする課題】しかしながら、上記従来の形態素解析方法には、以下のような問題点がある。

【００１２】上記の第１の従来の方法の最長一致法には、

（１）単語または文節を認識する段階で複数の候補が存在する場合に、入力字列の長さが長いものに高い優先順位を与えることの根拠が明らかではない：

（２）単語または文節の長さという局所的な評価しているために、文全体に対する尤度の評価ができない：という問題がある。

【００１３】さらに、上記の第２の従来の方法である文節数最小法は、

（１）入力文字列を構成する文節の総数が最小になる解釈を優先する根拠が明らかでない：

（２）一般に、文節数最小となる候補は複数存在するので、さらに何らかの手段で候補を詳細に順位付けする必要がある：等の問題がある。

【００１４】さらに、上記第３の従来の方法である接続コスト最小法は、具体的な接続コストの設定方法には言及しておらず、「自立語と付属語の接続コストは小さい」というような内容を表す接続コストが発見的な方法で与えられている。

【００１５】また、従来提案された確率正規文法に基づく日本語の形態素解析法には、次のような問題がある。

【００１６】（１）一般に、品詞の連接確率（品詞の二つ組の確率）だけでは、言語の局所的な性質を十分にモデル化できないことが多い。そこで、英語の確率的形態素解析では、品詞の三つ組の確率を用いることにより、精度の向上を達成している。しかし、接続コスト最小法及び確率文節文法、単語または品詞の接続（即ち、二つ組）に対してコスト与えられることを想定しており、三つ組の確率に基づくコストは扱えない。

【００１７】（２）接続コスト最小法は、最尤候補を１つだけ出力する。出力が上位Ｎ個の最尤候補を含むように、接続コスト最小法を拡張することは可能であるが、その場合には、予め、候補Ｎを決めておく必要がある。従って、例えば、形態素解析の出力を構文解析の入力として利用する場合などに、状況に応じて、最も尤もらしい候補から順番に任意の個数の候補を取り出すようなことはできない。

【００１８】また、上記の未知語の扱いの方法における前者の方法は、文字列の単語らしさを評価する方法であり、上記の後者の方法は、ある文脈における文字列の単語らしさを評価する方法とみなすことができるが、どちらの場合も尤度の根拠が不明確であり、単語仮説の詳細な順位付けも難しい。

【００１９】上記第６の従来の方法は、未知語の単語らしさを評価する有力な手段を与えるが、次のような問題点がある。

【００２０】（１）漢字で表記される複合語に対する造語モデルであり、それ以外では、造語単位の設定基準が必ずしも明確ではなく、自動的な処理が難しい。

【００２１】（２）生起確率の推定が目標であり、品詞

を考慮していない。従って、品詞の推定及び、品詞別の出力確率の推定ができない。

【００２２】（３）辞書の見出し語からモデルパラメータを推定するので、モデルパラメータが対象領域のテキストの性質を反映しない。

【００２３】（４）文字列の単語らしさを評価するだけで、その文字列が現れた文脈（前後の文字列）の情報を使って、単語としても尤もらしさを評価しているわけではない。

【００２４】本発明は、上記の点に鑑みなされたもので、上記従来の問題点を解決し、単語分割と品詞付与の組を、入力文が未知語を含む場合でも最も尤もらしい候補から順に任意の数だけ出力できるので、高い精度をもち、頑強でかつ柔軟なインタフェースを持つ日本語形態素解析装置及び日本語形態素解析方法を提供することを目的とする。

【００２５】

【課題を解決するための手段】図１は、本発明の原理構成図である。

【００２６】本発明の形態素解析装置は、日本語のの入力文が与えられ、形態素解析を行う装置であって、直前の２つの品詞が与えられた時の３つ目の品詞の確率である品詞三つ組確率と、品詞が与えられた時の単語の確率である品詞別単語出力確率から、文を構成する単語列と各単語に付与された品詞列の同時確率を与える品詞付けモデル１００と、単語に分割され、かつ、品詞が付与された文の集合から、品詞三つ組確率と品詞別単語出力確率を推定する品詞付けモデル推定手段１１０と、品詞三つ組確率を記憶する品詞三つ組確率テーブル１２０と、品詞別単語出力確率を記憶する品詞別単語出力確率テーブル１３０と　文頭からある単語に至るまでの単語列と品詞列の同時確率の最大値を記憶する最適経路スコアテーブル１７０と、最適経路スコアテーブル１７０に記憶されている文頭からある単語に至るまでの単語列と品詞列の同時確率の最大値を、単語を含む直前の２つの単語に付与された品詞の組が異なる場合毎に、単語を含む直前の３つの単語に付与された品詞三つ組確率、単語の品詞別単語出力確率、及び文頭からその単語の直前の単語に至るまでの単語列と品詞列の同時確率の最大値から求め、最適経路スコアテーブル１７０に記録する前向き探索手段１４０と、単語列と品詞列の同時確率を最大化するような入力文の単語分割及び品詞付与候補を、文末からある単語に至るまでの単語列と品詞列の同時確率の最大値、及び最適経路スコアテーブル１７０に記録された、文頭からその単語に至るまでの単語列と品詞列の同時確率の最大値から最も尤もらしい候補から順番に任意の個数だけ求める後向き探索手段１５０とを有し、品詞三つ組確率と品詞別単語出力確率から構成される品詞付けモデルに基づいて、まず、前向き探索手段１４０を用いて、文頭からある単語までの単語列と品詞列の同時確率の最大値を最適経路スコアテーブル１７０に記憶し、最適経路スコアテーブル１７０の値に基づいて後向き探索手段１５０を用いて最も尤もらしい順番に任意の個数の形態素解析候補を求める。

【００２７】また、本発明の日本語解析装置は、単語に分割され、かつ品詞が付与された文の集合から、品詞別文字三つ組の確率を推定する単語モデル推定手段２００と、品詞別文字三つ組の確率から、単語の品詞別出力確率を与える単語モデル１９０と、品詞別文字三つ組確率を記憶する品詞別文字三つ組確率テーブル２１０と、ある文字位置から始まる入力文の部分文字列の中から、品詞別文字三つ組確率に基づいて、最も尤もらしい順番に任意の個数の単語仮説、即ち、表記、品詞及び品詞別出力確率の組を求める単語仮説生成部１６０と、入力文中に未知語が存在する場合にも品詞別文字三つ組から構成される単語モデルを用いて、文字表記から単語仮説を生成し、後向き探索手段１５０において、文全体を考慮して最も尤もらしいように未知語の単語区切り、品詞及び品詞別単語出力確率を求める。

【００２８】図２は、本発明の原理を説明するためのフローチャートである。

【００２９】本発明の日本語形態素解析方法は、品詞三つ組確率と品詞別単語出力確率から求められる、文を構成する単語列と単語に付与された品詞列の同時確率を最大とするような形態素解析候補を、文頭から文末に探索する前向き探索と文末から文頭に探索する後向き探索を用いて、最も尤もらしい順番に任意の個数だけ求める。また、上記の前向き探索は動的計画法を用いて、文頭からある単語に至るまでの単語列と品詞列の同時確率の最大値を求め、記録する。

【００３０】また、上記の後向き探索はA*アルゴリズムを用いて、文末からある単語に至るまでの単語列と品詞列の同時確率の最大値と、前向き探索で記録された、文頭からある単語に至るまでの単語列と品詞列の同時確率の最大値により形態素解析候補の順位付けを行う。

【００３１】さらに、本発明は、入力文が辞書に登録されていない未知語を含む場合にあ、前向き探索において、単語を構成する文字列の始まりと終わり、品詞及び品詞別単語出力確率からなる単語仮説を、品詞別文字三つ組確率に基づいて入力文字列から生成し、後向き探索により、最も尤もらしい単語分割と品詞付与を決定することにより形態素解析候補を求める。

【００３２】

【作用】本発明は、シンボルの二次の拡大を考えて二重マルコフ過程を一重マルコフ過程に縮退させることにより、品詞三つ組確率を、従来の接続コスト最小法を拡張した枠組の中で扱えるようにする。具体的には、文頭から文末へ一文字ずつ進む動的計画法を用いて、単語列と品詞列の同時確率、即ち、品詞三つ組確率と品詞別単語出力確率の積を最大化するような形態素解析候補（単語

分割と品詞付与の組）を求める。この探索は、「前向き探索」と呼ばれる。前向き探索では、文頭からある文字位置までの入力文の部分列に対する単語系列と品詞系列の同時確率の最大値が、最適経路スコアテーブルに、最後の単語と最後の二重マルコフ過程の状態の異なる組み合わせ毎に記録される。

【００３３】次に、前向き探索で作成した最適経路スコアテーブルに基づいて、文末から文頭へ進むA*探索を用いて、最も尤もらしい順に一つずつ形態素解析候補を求める。この探索は、「後向き探索」と呼ばれる。後向き探索では、文末からある単語に至るまでの部分経路（単語列と品詞列の組）の候補が完全な経路のコスト、即ち、後向きの部分経路のコストと残りの最適部分経路のコストの和、に基づいて順位付けされる。残りの最適部分経路、即ち、その単語から文頭に至るまでの最適部分経路のコストは、前向き探索において、最適経路スコアテーブルに記録されているので、A*探索の性質により、この方法で必ず最適解が求められることが保証されている。また、最適解が求められた後に、この解を取り除き、さらに、A*探索を続けることにより、次の最適解を求められることも保証されている。

【００３４】従って、このような、マルコフ過程の縮退、前向きの動的計画法、及び後向きのA*探索の組み合わせにより、（１）品詞の三つ組の確率を用いて、（２）最も尤もらしい順に任意の個数の候補を求める、日本語形態素解析装置が実現できる。また、この方法は、任意の高次のマルコフ過程を扱うように、自然に拡張できる。

【００３５】未知語を推定する問題は、（１）単語境界（単語を構成する文字列の始まりと終わり）、（２）品詞別の出力確率、からなる三つ組を求める問題に帰着される。この三つ組のことを「単語仮説」と呼ぶことにする。入力文字列から辞書に収録されていない単語仮説を生成する手段があれば、入力文が未知語を含む場合でも、文全体を考慮して最も尤もらしい形態素解析候補を求めることができる。

【００３６】単語を構成する文字列と品詞が与えられた時に、その単語の品詞別の出力確率を計算するモデルのことを「単語モデル」と呼ぶことにする。単語モデルがあれば、入力文の任意の部分文字列の単語らしさを判定することができる。

【００３７】本発明では、品詞別文字三つ組確率から構成される単語モデルを用いて、入力文に対して単語仮説を生成する。なお、単語モデルのパラメータは、品詞タグ付コーパスから推定する。具体的には、前向き探索では、入力文の全ての文字位置において、その最左部分文字列と一致し、かつ、辞書に登録された単語を提案する。これに加えて、未知語を扱うために、最左部分文字列と一致するが、辞書には登録されていない単語仮説を、予め決めた上限以下の個数、最も尤もらしいものから順に、提案する。次に、後向き探索では、文全体を考慮に入れて最も尤もらしい単語分割と品詞付与を決定する。

【００３８】従って、このような、統計的な単語モデルに基づく文字表記からの単語仮説生成法、及び、上記前向き探索と後向き探索から構成される日本語形態素解析法の組み合わせにより、（１）未知語の単語らしさを自動的に評価でき、（２）品詞推定が可能で、（３）対象とするテキストデータの性質を反映し、（４）文全体から見ても最も妥当な解釈をする、日本語形態素解析装置が実現できる。

【００３９】

【実施例】以下、図面と共に本発明の実施例を詳細に説明する。

【００４０】図３は、本発明の一実施例の概略ブロック図を示す。同図に示すように、本発明の形態素解析装置は、前向きＤＰ探索部１、後向きA*探索部２、品詞タグ付きコーパス３、品詞付けモデル推定部４、品詞三つ組確率テーブル５、品詞別単語出力確率テーブル６、単語モデル推定部７、品詞別文字三つ組確率テーブル８、単語仮説生成部９及び、最適経路スコアテーブル１０より構成される。

【００４１】前向きＤＰ探索部１は、入力文の文頭から文末へ一文字ずつ進む動的計画法（DP:Dynamic Programing)を用いて、単語列と品詞列の同時確率、即ち、品詞三つ組確率と品詞別単語出力確率の積を最大化するような、入力文の単語分割と品詞付与の組を求める。品詞２つ組確率テーブル５及び品詞別単語出力確率テーブル６は、それぞれ、品詞三つ組確率及び品詞別単語出力確率を格納する。この前向きの動的計画法では、文頭からある単語に至るまでの単語列と品詞列の同時確率を最大とする最適部分経路（単語列と品詞列の組）の確率が、最後の単語と最後の二重マルコフ過程の状態の異なる組み合わせ毎に計算され、最適経路スコアテーブル１０に記録される。

【００４２】単語仮説生成部９は、前向きの動的計画法において、入力文のある文字位置における最左部分列と照合する単語を検索する際に、最左部分列を単語表記とする単語仮説を最も尤もらしい順に、予め定めた個数以下だけ生成する。

【００４３】後向きA*探索部２は、前向きＤＰ探索部１で計算された最適経路スコアテーブル１０の内容を入力とし、文末から文頭へ一単語ずつ進むA*アルゴリズムを用いて、品詞三つ組確率と品詞別単語出力確率の積が最も大きいものから順番に一つずつ形態素解析候補を求める。

【００４４】品詞付けモデル推定部４は、品詞タグ付きコーパス３、即ち、単語に分割され、かつ、品詞が付与された大量の日本語文のデータから、品詞三つ組確率と品詞別単語出力確率を推定し、品詞三つ組確率テーブル

５及び品詞別単語出力確率テーブル６へ格納する。

【００４５】単語モデル推定装置７は、品詞タグ付きコーパスから品詞別文字三つ組確率を推定し、品詞別文字三つ組確率テーブル８へ格納する。

【００４６】以下では、品詞付けモデルの推定、マルコフ過程の縮退、動的計画法を用いた前向き探索、A*アルゴリズムを用いた後向き探索、単語モデルの推定、単語仮説の生成の順に説明する。

【００４７】（１）　品詞付けモデルの推定

品詞付けモデル推定部４が品詞タグ付きコーパスから、品詞三つ組確率と品詞別単語出力確率から構成される品詞付けモデルを推定する手順を示す。

【００４８】本発明では、日本語の形態素解析のために、品詞三つ組モデル（トライポスモデル（tri-POS））と呼ばれる統計的言語モデルを用いる。

【００４９】まず、入力文が単語系列$W=w_1, w_2, \cdots w_n$に分割され、各単語に品詞系列$T=t_1, t_2, \cdots t_n$が付与されたとする。トライポス（以下tri-POS）モデルでは、式(1)に示すように、単語列と品詞列の同時確率$P(W, T)$を、品詞三つ組確率$P(t_i \mid t_{i-2}\, t_{i-1})$と品詞別単語出力$P(w_i \mid t_i)$の積で近似する。従って、日本語の形態素解析は、品詞列と単語列の同時確率を最大化する単語分割と品詞付与の組を見つける問題に帰着される。

【００５０】

【数１】

$$P(W, T) = \prod_{i=3}^{n} P(t_i \mid t_{i-2}, t_{i-1})\, P(w_i \mid t_i) \qquad (1)$$

【００５１】実用的には、文の境界が重要な情報を担っているので、式(1)の代わりに式(2)を用いるのが有効である。ここで"#"は文頭及び文末を表す特別な記号である。

【００５２】

【数２】

$$P(W, T) = P(t_1 \mid \#)\, P(w_1 \mid t_1)\, P(t_2 \mid \#, t_1)\, P(w_2 \mid t_2)$$

$$\prod_{i=3}^{n} P(t_i \mid t_{i-2}, t_{i-1})\, P(w_i \mid t_i)\, P(\# \mid t_{n-1}, t_n) \qquad (2)$$

【００５３】品詞三つ組確率$P(t_i \mid t_{i-2}, t_{i-1})$と品詞別単語出力確率$P(w_i \mid t_i)$は、次式(3)により品詞タグ付きコーパスにおける相対頻度から推定できる。但し、fは相対頻度を求める関数、$N(w, t)$は品詞タグtを持つ単語wが現れた回数、$N(t_{i-2}, t_{i-1}, t_i)$は、品詞タグ列t_{i-2}, t_{i-1}, t_iが現れた回数である。

【００５４】

【数３】

$$P(t_i \mid t_{i-2}, t_{i-1}) = f(t_i \mid t_{i-2}, t_{i-1}) = \frac{N(t_{i-2}, t_{i-1}, t_i)}{N(t_{i-2}, t_{i-1})} \qquad (3)$$

$$P(w_i \mid t_i) = f(w_i \mid t_i) = \frac{N(w, t)}{N(t)} \qquad (4)$$

【００５５】一般に、相対頻度から、直接、品詞三つ組確率を推定するのに十分な量の品詞タグ付きコーパスを用意するのは難しい。そこで、品詞三つ組確率の平滑化のために、内挿推定法を用いる。これは、式(5)に示すように、一つ組、二つ組、三つ組の相対頻度を内挿することにより、三つ組の確率を求める方法である。

【００５６】

$$P(t_i \mid t_{i-2}, t_{i-1}) = q_3 f(t_i \mid t_{i-2}, t_{i-1}) + q_2 f(t_i \mid t_{i-1}) + q_1 f(t_i) + q_0 V \qquad (5)$$

ここで、fは相対頻度を求める関数であり、Vはすべての品詞タグが一様に出現する確率である。非負の重み計数q_iは$q_3+q_2+q_1+q_0=1$を満足する。内挿推定法では、観測データの確率が最大となるように、重

み計数が決定される。

【００５７】図４は、品詞タグ付きコーパスの一例を示す。この例では、各文に対して文識別番号が与えられ、各文は、空白文字により単語に区切られている。各単語は、その表記に続いて品詞が付与されている。ここで、“／”は表記の品詞の区切りを示す記号である。例えば、『もしもし通訳電話国際会議事務局ですか？』については、

“3001-100

もしもし/ 感動詞、/ 記号　通訳電話国際会議事務局/ 固有名詞　です/ 助動詞・終止　か/ 終助詞　?/記号”

となる。

【００５８】図５は、品詞三つ組確率の一例である。品詞三つ組確率のデータは三つの品詞とその確率の４つの要素から構成されるリスト構造で表されている。

【００５９】図６は、品詞別単語出力確率の一例である。品詞別単語出力確率のデータは、品詞、表記、確率の３つの要素から構成されるリスト構造で表される。

【００６０】（２）マルコフの過程の縮退

以下では、前向きＤＰ探索と後向きA*探索を用いてN-Bestの形態素解析候補を求めるアルゴリズムが、式(1)で表される品詞モデルに対して、二次マルコフ過程を一次マルコフ過程に縮退させる操作を施すことにより導かれることを示す。

【００６１】$u_i = t_i$及び$u_i = t_{i-1}\ t_i$とすることにより、結合状態系列$U = u_1, u_2, \cdots, u_n$を定義すると、次の関係が成り立つ。

【００６２】

$$P(u_i \mid u_{i-1}) = P(t_i \mid t_{i-2}, t_{i-1}) \qquad (6)$$

式(6)を式(1)に代入すると、式(7)が得られる。

【００６３】

【数４】

$$P(W, T) = \prod_{i=1}^{n} P(u_i \mid u_{i-1})\, P(w_i \mid t_i) \qquad (8)$$

【００６４】式(7)は、通常の一次マルコフ過程と同じ形式である。このように、結合状態を考えることにより、任意の高次マルコフ過程は、一次マルコフ過程へ還元することができる。部分単語系列を$W_i = w_1, w_2, \cdots, w_n$、部分品詞系列を$T_i = t_1 \cdots t_i$と定義すると、次のような関係がある。

$$P(W_i, T_i) = P(W_{i-1}, T_{i-1})\, P(u_i \mid u_{i-1})\, P(w_i \mid t_i) \qquad (8)$$

この式(8)から、各結合状態u_iに対して単語列と品詞列の同時確率$P(W_i, T_i)$の最大値を求めるには、(1) 各結合状態u_{i-1}に対して単語列と品詞列の同時確率$P(W_{i-1}, T_{i-1})$の最大値を記憶し、(2) 各結合状態u_{i-1}に対する同時確率$P(W_{i-1}, T_{i-1})$の最大値と式(8)を用いて全ての結合状態系列u_iと全ての部分単語系列w_iについて同時確率$P(W_i, T_i)$を計算し、各結合状態系列u_iに対する同時確率$P(W_i, T_i)$の最大値を選択すればよいことがわかる。

【００６５】従って、ｉを１からｎまで１つずつ増やすことにより、同時確率$P(W_n, T_n)$を最大化する結合状態系列u_nを求めることができる。しかし、日本語は単語を分かち書きする習慣がないので、入力文の単語分割を事前に一通りには決めることができない。そこで、前向き探索は、動的計画法を用いて、様々な単語分割の可能性を考慮しながら、同時確率$P(W_n, T_n)$を最大化する結合状態系列u_nを求められるように工夫されている。

【００６６】次に、同時確率$P(W_n, T_n)$を最大化する結合状態系列u_nが求まれば通常のビタビアルゴリズムと同様に、この結合状態系列u_nへ至るまでの経路を逆に辿ることにより、同時確率$P(W_n, T_n)$を最大化する結合状態系列$U = u_1, u_2 \cdots u_n$と単語系列$W = w_1 \cdots w_n$が得られる。この結合状態系列において、各結合状態の最初の状態を無視すれば、同時確率$P(W_n, T_n)$を最大化する状態系列（品詞タグ系列）$T = t_1 \cdots t_n$が得られる。これが、後向き探索である。後向き探索では、前向き探索で求めた部分形態素列の確率の最大値をA*アルゴリズムのヒューリスティック関数として用いることにより、最も尤もらしい順に一つずつ形態素解析候補を取り出すことができる。

【００６７】（３）動的計画法を用いた前向き探索

前向きＤＰ探索部１が最適経路スコアテーブル１０に格納する値を計算する手順を示す。最初に前向きＤＰ探索部で用いられるデータ構造及び、補助的な関数について説明し、続いて具体的な処理の流れを説明する。

【００６８】まず、図７に示すようなスロット（フィールド）を持つ、“parse ”と“word”という２つの構造体（レコード）を定義する。構造体parse は、最適経過スコアテーブルにおいて、単語の情報及び文頭からその単語へ至る最適部分経路（同時確率が最大となるような部分単語列と部分品詞列の組、即ち形態素列）の情報を格納するのに用いられる。パーススタート“parse. start”とパースエンド“parse. end”は、入力文における単語の開始位置と、終了位置のインデックスである。

“parse.pos ”は、単語の品詞で、ここでは、単語の品詞、活用型、活用形のリストを用いている。“Parse.nt

h-order-state ”は、この単語を含む最後の2 つの単語の品詞のリストである。このスロットは、二重マルコフ過程の結合状態に対応する。“Parse.prob-so-far ”は、文頭から現在の単語に至るまでの最適部分経路のスコアである。構造体wordは、辞書(品詞別単語出力確率テーブル６）において、個々の単語の情報を格納するのに用いられる。“Word.form, word.pos,word.prob ”は、それぞれ、単語の表記、品詞、品詞別の出力確率を表す。

【００６９】最適経路スコアテーブル(以下ではパーステーブルと呼ぶ) は、最後の単語と最後の二重マルコフ過程の状態、即ち、“parse.start, parse. end , parse.nth-order-state ”の値のリストをキーとし、同じキーを持つパース(parse) 構造の中で、最適部分経路スコアが最良なもの(複数ある場合はそのリスト) を値として保持するテーブルである。

【００７０】このパーステーブルの操作関数として、レジスタパース(resister-parse)とゲットパース(get-parses)を定義する。関数レジスタパース(resister-parse)はパース構造と最適経路スコアテーブルを引数として、そのパース構造をパーステーブルに登録する。この場合、同じキーを持つパース構造が既に登録されている場合は、最適部分経路のスコアが同じならば、両方ともテーブルに残す。関数ゲットパース(get-parses)は、入力文の文字位置と最適経路スコアテーブルを引数とし、その文字位置が単語の終了位置となっている。パーステーブルの中の全ての要素(パース構造) を集め、最後の二つの品詞(parse.nth-order-state) が同じパース構造の中で、最良の最適部分の経路のスコア(parse.prob-so-far) を持つパース構造のみを集めたのを返す。

【００７１】図８は、本発明の一実施例の前向きＤＰ探索部の動作を説明するたのフローチャートである。以下では、図７に従って前向きＤＰ探索部１の動作を説明する。まず、パーステーブルには、文頭に対応する特別なパース構造を予め登録し、入力文字列の最後には、文末に対応する特別な記号を付加する。

【００７２】前向きＤＰ探索は、入力文の先頭から始まり、文末方向へ一文字ずつ進む。

【００７３】ステップ１）入力文の文字位置を表す変数ｉを０に設定する。

【００７４】ｉ＝０

ステップ２）探索が文末に達したかを判断する。ここで、関数lengthは、文字列の長さを返す関数である。もし、文末に達していれば、前向きＤＰ探索を終了する。そうでなければ以下の処理を各文字位置で行う。

【００７５】ｉ＜lenth (string)

ステップ３）まず、関数get-parse を用いて単語の終了位置が文字位置ｉであるパース構造の中で、最後の二つの単語の品詞の異なる組み合わせごとに、最良の部分経路スコアを持つものからなるリストが求められる。次に、その先頭要素が変数parse に、残りの要素が変数parsesに格納される。ここで、関数first はリストの先頭要素を返し、関数restはリストから先頭要素を除いたリストを返すものとする。

```
parses:=get-parses(i,parse-table);
parse:=first(parses);
parses:=rest(parses);
```

ステップ４）最後の２つの単語の品詞の異なる組み合わせ毎に、最適部分経路スコアを持つパース構造のリストの終わりに達したかどうかを判断する。もし、そうなら、ステップ１２において文字位置のインデックスをインクリメントする。そうでなければ、ステップ５以降の処理を各パース構造に対して行う。

【００７６】parse==nil ？

ステップ５）関数get-leftmost-substrings を用いて文字位置ｉから始まる部分列と一致する辞書中の単語のリストが求められ、変数words に代入される。次にその先頭要素が変数wordに、残りの要素が変数words に代入される。

```
words:=get-leftmost-substrings(string,i);
word:=first(words);
words:=rest(words);
```

ステップ６）文字位置ｉから始まる部分列と一致するすべての単語が調べられたかどうかを検査する。もし全て調べたのであれば、ステップ１１においてparsesの先頭要素をparse に代入し、parsesの先頭要素を取り除く。そうでなければ、ステップ７以降の処理を行う。

【００７７】word==nil ？

ステップ７）変数parse,nth-order-state の値と変数wordのpos スロットの値から、品詞の三つ組を表すリストを作成し、変数pos-ngram に代入する。ここで、関数listは引数を要素とするリストを返し、関数appendは引数のリストに含まれる要素を連結したリストを返すものとする。

【００７８】pos-ngram:=append(pase.nth-order-state,list(word.pos);

ステップ８）変数pos-ngram に代入された品詞三つ組の確率が調べられる。もし、品詞三つ組確率が０ならば、ステップ１０において、次の要素を変数wordへ代入する。もし、品詞三つ組確率が０でなければ、以下の処理を行う。ここで、関数transprob は引数の品詞三つ組の確率を返すものとする。

【００７９】transprob(pos-ngram)=0

ステップ９）まず、新しいパース構造をつくり変数new-parse に代入する。関数make-parseは新しいパース構造を返すものとする。この新しいparse 構造の開始位置(new-parse.start) は、ｉ、終了位置 (new-parse.end) は、ｉと変数wordに格納された単語の表記の長さの和、品詞(new-parse.pos) は、変数wordに格納された単語の品詞である。また、結合状態(new-parse.nth-order-sta

te）は、品詞三つ組の先頭要素を取り除いたものであり、最適経路スコア(new-parse.prob-so-far）は、直前の単語の最適経路スコア(parse.prob-so-far）と品詞三つ組確率(transprob(pos-ngram))と単語出力確率(word.prob）の積である。次に、この新しいparse 構造を、関数register-parseを用いてパーステーブル(parse-table）に登録し、ステップ１０に移行する。

```
new-parse:=make-parse();
new-parse.start:=i;
new-parse.end:=i+length(word.form);
new-parse.pos:=word.pos;
new-parse.nth-order-state:=rest(pos-ngram);
new-parse.prob-so-far:=parse.prob-so-far* transpro
b(pos-ngram)* wrod.prob;
register-parse(new-parse,parse-table);
```

ステップ１０）リストwords の先頭要素をwordへ代入し、words の残りの要素をwords へ代入する。

```
word:=first(words);
words:=rest(words);
```

ステップ１１）リストparsesの先頭要素をparse へ代入し、parsesの残りの要素をparsesへ代入する。

```
parse:=first(words);
parses:=rest(parses);
```

本発明は、後向き探索では、関数ｇとして文末から現在の単語（パース構造)に至るまでの品詞三つ組確率と品詞別単語出力確率の積の対数の絶対値を用いる。また、関数h としては、文頭から現在に至るまでの品詞三つ組確率と品詞別単語出力確率の積の最大値の対数の絶対値を用いる。

【００８４】この後向きA* 探索のために図６に示すようなスロット(フィールド）を持つパス(path)という構造体を定義する。構造体pathはA* 探索におけるグラムのノードに相当し、現在の単語(parse構造）、これまでの経路及びコストに関する情報を保持する。Path.parse は、parse 構造を格納する。Path.previous は直前のpath構造へのポインタである。Path.cost-so-farは、初期状態からのコストである。Path.total-cost は、初期状態から最終状態までのコストの推定値である。

【００８５】A* 探索では、“open”と“close ”という二つのリストを用いる。リスト“open”は、既に生成され、ヒューリスティック関数が適用されているが、まだ展開されて(調べられて）いないノード(path 構造）の集合である。このリストはヒューリスティック関数の値に基づく優先度付きキュー(priority queue)になっている。リスト“close ”は、既に展開された(調べられた）ノードの集合である。

【００８６】A* 探索では、目標状態に対応するノードを生成するまで、各ステップの一つのノードを展開する。各ステップでは、既に生成されているが、まだ、展

ステップ１２）変数ｉの値をインクリメントする。

【００８０】ｉ＝ｉ＋１

（４）A* アルゴリズムを用いた後向き探索

後向きA* 探索部２が最も尤もらしい順にひとつずつ形態素解析候補を求める手順を示す。最初に、A* 探索の概要、後向きA* 探索部で用いられるデータ構造、及び、補助的な関数について説明し、続いて、具体的な処理の流れを説明する。

【００８１】本発明の後向きA* 探索では、単語と品詞の組であるパース構造を、A* アルゴリズムにおけるグラフのノードと考える。そして、コストとしては、確率の対数の絶対値を用いる。これにより、確率最大の解は、コスト最小の解に対応し、確率の積はコストの和に対応する。

【００８２】A* 探索では、ヒューリスティック関数ｆ（ｎ）を考える。ヒューリスティック関数ｆ（ｎ）は、現在のノードｎを生成した経路に沿って、初期状態から最終状態へ至るまでのコストの推定値を与える。初期状態から現在のノードへ至るまでのコストを与える関数をｇ（ｎ）、現在のノードから最終状態へ至るまでのコストの推定値を与える関数をｈ（ｎ）とすると、ヒューリスティック関数ｆ（ｎ）は次式により与えられる。

【００８３】

$$f(n)=g(n)+h(n) \qquad (9)$$

開されていない最も有望なノードを展開する。即ち、選ばれたノードの後続のノード(successors)を生成し、ヒューリスティック関数を適用し、既に生成されていないかを検査した後にリスト“open”に加える。この検査によって、各ノードはグラフの中に一回だけ現れることが保証される。また、二つ以上の経路が同じノードに生成する時は、スコアの良い方を記録する。

【００８７】リスト“open”とリスト“close ”を操作する補助関数として、“find-path, insert-and-sort-path, delete-path ”を定義する。関数“find-path ”はパース構造とリスト(open 又はclose)を引数とし、引数のパース構造と、次に定義する意味で「等しい」パース(parse）構造を持つパス(path)構造がリスト中に存在すれば、それを返す。ここで、二つのパース構造の開始位置と終了位置と結合状態が等しい場合、この二つのパース構造は「等しい」と定義する。

【００８８】関数“insert-and-sort-path”はパス(path)構造とリストを引数とし、リストにパスを挿入し、初期状態から最終状態までのコストの推定値“path.total-cost ”の小さい順にソートする。関数“delete-path ”は、パス構造とリストを引数とし、引数のパス構造をリストから削除する。

【００８９】また、後向き探索において、ノード（パス構造）の後続ノードを生成するための補助関数として、“immediate-left-parses ”を定義する。関数“immediate-left-parses ”は、パース構造とパーステーブル(

最適経路スコアテーブル）を引数とし、引数のパース構造の左側（文頭側）に連接しうるすべてのパース構造のリストを返す。具体的には、次に三つの条件を満たすパース構造をパーステーブルの中から検索する。

（１）引数のパース構造の開始位置がこのパース構造の終了位置である：

（２）引数のパース構造の結合状態のリストか最後の要素を無視したものと等しい：

（３）このパース構造から引数のパース構造への前向きの状態遷移を表す品詞三つ組の確率が０ではない：さらに、後向き探索において、初期状態から現在のノードへ至るまでのコストを計算するために、ノード間の遷移コストを計算する関数として“transition-and-word-cost”を定義する。関数“transition-and-word-cost”は、隣接する二つのパース構造を引数（第１引数が文末側の単語で、第２引数が文頭側の単語）とし、文末側の単語の品詞別単語出力確率の対数と品詞三つ組確率の対数の和の絶対値を返す。

【００９０】また、現在のノードから最終状態へ至るまでのコストを得るために、関数“cost-from-beginning-of-sentence”を定義する。関数“cost-from-beginning-of-sentence”は、文頭から引数のパース構造に至るまでのコストを返す。このコストは、既に前向き探索で既に計算されており、“parse.prob-so-far”に格納されている。

【００９１】図９は、本発明の一実施例のA*探索部の動作を説明するためのフローチャートである。以下では、同図に従って、後向きA*探索部の動作を説明する。

【００９２】ステップ１０１）文末において、最適経路コストをもつパース構造をパーススロットに持つパス構造からなるリストを“open”に代入する。ここで、関数“backward-search-initial-paths”は、このような初期化を行う関数とする。また、“closed”には、空リストを代入する。

open:=backward-serch-initial-paths();

closed:=nil;

ステップ１０２）“open”が空リストかどうかを調べる。もし空リストであれば、解がみつからなかったので探索が失敗したことを通知して探索を終了する。そうでなければ、ステップ１０３以降の処理を行う。

【００９３】open==nil ？

ステップ１０３）“open”の先頭要素を変数“bestpath”に代入する。

【００９４】bestpath:=first(open)

ステップ１０４）探索が文頭に達したかを調べる。ここで、関数“is-beginning-of sentence”は、この条件を調べる関数である。もし、探索が文頭に達していれば、“bestpath”が最適解であり、探索が成功したことを通知して探索を終了する。そうでなければ、ステップ１０５以降の処理を行う。また、探索は成功したが、さらにその次に最も尤もらしい解を求めたい場合にも、ステップ１０５以降の処理を行う。

【００９５】

is-beginning-of-sentence(bestpath-parse) ？

ステップ１０５）まず、変数“open”へ“open”から先頭要素を取り除いたリストを代入する。次に、関数“insert-and-sort-path”を用いて、“bestpath”をリスト“closed”へ挿入し、“closed”の要素を初期状態から最終状態までのコストの推定値に順にソートする。

【００９６】open:=rest(open);

insert-and-sort-path(bestpath, closed)

ステップ１０６）関数“immediate-left-parses”を用いて、“bestpath”のパーススロットに格納されたパース構造の左側に連接するすべてのパース構造をパーステーブルから取り出したリストを作り、このリストを“successorparses”に代入する。続いて、successor-parses”の先頭要素を“successorparse”に代入し、残りを“successorparses”に代入する。

【００９７】successorparses:=immidiate-left-parses(bestpath.parse, parse-table);

successorparse:=first(successorparses);

successorparses:=rest(successorparses);

ステップ１０７）“successorparses”のすべての要素を処理したかを検査する。もし、そうであれば、ステップ１０２に移行し、そうでなければステップ１０８の処理を行う。

【００９８】succesorparse==nil ？

ステップ１０８）新しいパス構造を割り当て、これを変数“newpath”に代入する。関数“make-path”は新しいパス構造のための記憶領域を割り当てる関数である。そして、“newpath.parse”には、“successorparse”を代入し、“newpath.previous”には“bestpath”を代入する。“newpath.cost-so-far”には、“bestpath.parse”へ至るまでのコスト“bestpath.cost-so-far”と、“bestpath.parse”から“newpath.parse”への遷移のコストの和が代入される。この遷移コストは、関数“transition-and-word-cost”を用いて計算される。“newpath.total-cost”には、文末から“newpath.parse”までのコスト(newpath.cost-so-far)と、文頭からnewpath.parseまでのコストの和が代入される。文頭からのコストは関数“cost-from-beginning-of-sentence”を用いて計算される。

newpath:=make-path();

newpath.parse:=successorparse;

newpath.previous:=bestpath;

newpath.cost-so-far:=bestpath.cost-so-far

+transition-and-word-cost(bestpath.parse, newpath, parse);

newpath.total-cost-so-far

+cost-from-beginning-of-sentence(newpath.parse)；
ステップ１０９）関数“find-path ”を用いて、successorparsse をパーススロットにもつパス構造が“open”に含まれているかを検査する。もし含まれていなければ、ステップ１１３に移行する。含まれていれば、ステップ１１０以降の処理を行う。
【００９９】find-path(successorparse,open)==nil ？
ステップ１１０）“succdessorparse ”をパーススロットに持つ“open”の中のパス構造を変数“oldpath ”に代入する。
【０１００】
oldpath:=find-path(successorparse,open)；
ステップ１１１）“newpath.total-cost”と“oldpath.total-cost”を比較する。もし、“newpath.total-cost”の方が大きければ、何もせずにステップ１１８に移行する。もし、“newpath.total-cost”の方が小さければ、ステップ１１２に移行する。
【０１０１】
newpath.total-cost＜oldpath.total-cost ？
ステップ１１２）関数“delete-path ”を用いて“open”から“oldpath ”を削除し、関数“insert-and-sort-path”を用いて、“newpath ”を“open”へ挿入した後にソートする。そしてステップ１１８に移行する。
【０１０２】delete-path(oldpath.open)；
insert-and-sort-path(newpath,open)；
ステップ１１３）関数find-path を用いてsuccessorparseをパススロットに持つパス構造が“closed”に含まれているかを検査する。もし含まれていれば、ステップ１１７に移行する。含まれていればステップ１１４以降の処理を行う。
【０１０３】
find-path(successorparse,closed)==nil ？
ステップ１１４）“successorparse”をパーススロットに持つ“closed”の中のパス構造を変数“oldpath ”に代入する。
【０１０４】
oldpath:=find-path(successorparse,closed)；
ステップ１１５）“newpath.total-cost”と“oldpath.total-cost”を比較する。もし、“newpath.total-cost”の方が大きければ、何もせずにステップ１１８に移行する。
【０１０５】newpath.total-cost＜oldpath.total-cost
ステップ１１６）ステップ１１５においてもし、“newpath.total-cost”の方が小さければ、関数“delete-path ”を用いて、“closed”から“oldpath ”を削除、関数“insert-and-sort-path”を用いて“newpath ”を“closed”へ挿入した後にソートする。そしてステップ１１８に移行する。
delete-path(oldpath,closed)；
insert-and-sort-path(newpath,open)
ステップ１１７）関数“insert-and-sort-path”を用いて“newpath ”を“open”へ挿入した後にソートする。そしてステップ１１８に移行する。
【０１０６】inset-and-sort-path(newpath,open)；
ステップ１１８）“successorparses ”の先頭要素を“successorparse”へ代入し、“successorparses ”の残りの要素を“successorparses ”へ代入する。そして、ステップ１０７に移行する。
successorparse:=first(successorparses)：
successorparses:=rest(successorparses)：
図３の後向きA*探索部２により得られた形態素解析候補の例において、図１０の各文に対して上記３個の形態素解析候補が示されている。各形態素解析候補には、確率の対数が示されており、この値が大きいほど尤もらしい。
【０１０７】図１０において、『会議に申し込みたいのですが。』が入力文として与えられたときに、第１候補の確率が自然対数をとったものが、-25.82409083887518であり、この形態素解析候補は、「会議」「に」「申し込み」「たい」「の」「です」「が」「。」といいう単語系列と、「普通名詞」「格助詞」「本助詞・連用・五段」「助動詞・連体」「準体助詞」「助動詞・終止」「接続助詞」「記号」という品詞系列から構成されることを表している。第２候補及び第３候補についても同様である。確率は、自然対数をとっているので、負の値となるが、値が大きいほど（絶対値が小さいほど）尤もらしい候補となる。
【０１０８】（５）単語モデルの推定
単語モデル推定部７が、品詞タグ付きコーパスから品詞別文字三つ組確率から構成される単語モデルを推定する手順を示す。
【０１０９】本発明では、品詞別の文字三つ組確率を用いて、単語モデルを作成する。単語ｗを構成する文字列を$C=c_1 c_2 \cdots c_n$とする。品詞ｔが与えられた時の単語の出力確率を次式で近似させる。ここで“#”は単語境界を示す。
【０１１０】
【数５】

$$P(w \mid t) = P_t(C) = P_t(c_1 \mid \#, \#)\, P_t(c_2 \mid \#, c_1) \prod_{i=3}^{n} P_t(c_i \mid c_{i-2}, c_{i-1})\, P_t(\# \mid c_{n-1}, c_n) \tag{10}$$

【０１１１】品詞別文字三つ組確率は、品詞タグ付きコーパスにおいて、品詞がｔである全ての単語に現れた文字三つ組と文字二つ組の相対頻度か次式により推定できる。ここで、$N_t(c_{i-2}, c_{i-1}, c_i)$は、品詞タグｔが与えられた単語に現れた文字三つ組$c_{i-2}$，$c_{i-1}$，$c_i$の総数である。この相対頻度は、単語の出現頻度を反映することに注意して欲しい。

【０１１２】

【数６】

$$P_t(c_i \mid c_{i-2}, c_{i-1}) = f_t(c_i \mid c_{i-2}, c_{i-1}) = \frac{N_t(c_{i-2}, c_{i-1}, c_i)}{N_t(c_{i-2}, c_{i-1})} \tag{11}$$

【０１１３】一般に、この文字三つ組確率を推定するのに十分な量の訓練データを用意することは難しい。そこで、品詞三つ組の場合と同様に、次式に示すように内挿推定法を用いて文字三つ組確率を平滑化する。

$$P(c_i \mid c_{i-2}, c_{i-1}) = q_3 f(c_i \mid c_{i-2}, c_{i-1}) + q_2 f(c_i \mid c_{i-1}) + q_1 f(c_i) + q_0 V \tag{12}$$

各品詞別単語モデルは、ある文字列がその品詞の単語である場合の単語出力確率を求める。従って、単語仮説の文字列表記と品詞が与えられれば、この単語モデルから品詞別単語出力確率を求めることができる。なお、品詞別単語モデル（文字三つ組モデル）は、名詞や動詞等の全てのオープンカテゴリ、即ち、その品詞に属する単語の数が有限ではないと考えられる品詞に対して作成する。

【０１１４】（６）単語仮説の生成

単語仮説生成部９が、単語モデルを用いて入力文に対して単語仮説を生成する手順を説明する。

【０１１５】単語仮説生成部９は、未知語を扱うために、前向き探索において、入力文のある文字位置から始まる任意の長さの部分文字列と全てのオープンカテゴリ（品詞）の組に対して、単語モデルを用いて品詞別単語出力確率を計算し、最も尤もらしい順に予め決められた個数の単語仮説を生成する。

【０１１６】図１１は、単語仮説生成部の動作を説明するためのフローチャートである。以下では、同図に従って、単語仮説生成部９の動作を説明する。

【０１１７】単語仮説生成部９は、入力文字列“string”、及び単語仮説を生成すべき文字位置ｉを引数として与えられる。また、定数“delimiters”には、句点（。）や読点（、）等の単語の一部とはなり得ない区切り記号のリストが格納され、定数“open-categories”には、名詞や動詞などのその品詞に属する単語の数が有限ではない品詞（オープンカテゴリ）のリストが格納されている。

【０１１８】ステップ２０１）単語仮説のリストを格納する変数“word-hypos”に初期値として“nil”を代入する。

【０１１９】word-hypos:=nil;

ステップ２０２）単語仮説の長さを格納する変数ｊに初期値として０を代入する。

【０１２０】ｊ：＝０

ステップ２０３）単語仮説の長さｉが単語仮説を生成すべき文字位置ｉから文末までの文字列の長さより大きくなったかどうかを調べる。もし、そうならば、ステップ２０９に移行する。そうでなければ、ステップ２０４以降の処理を行う。

【０１２１】ｊ＞length(string)-i

ステップ２０４）単語仮説となる文字列の最後の文字が区切り記号であるかどうかを調べる。もし、そうならば、ステップ２０９に移行する。区切り記号でなければ、ステップ２０５以降の処理を行う。

【０１２２】member(char(string,i+j-1),delimiters)

ステップ２０５）入力文字列“string”の位置ｉから始まる長さｊの部分文字列を変数formへ代入する。ここで、関数substrは、部分列を返す関数とする。

【０１２３】form:=substr(string,i,j);

ステップ２０６）オープンカテゴリリスト“open-categories”の先頭要素を変数pos に代入し、残りを変数“poses”に代入する。

【０１２４】poses:=open-categories;

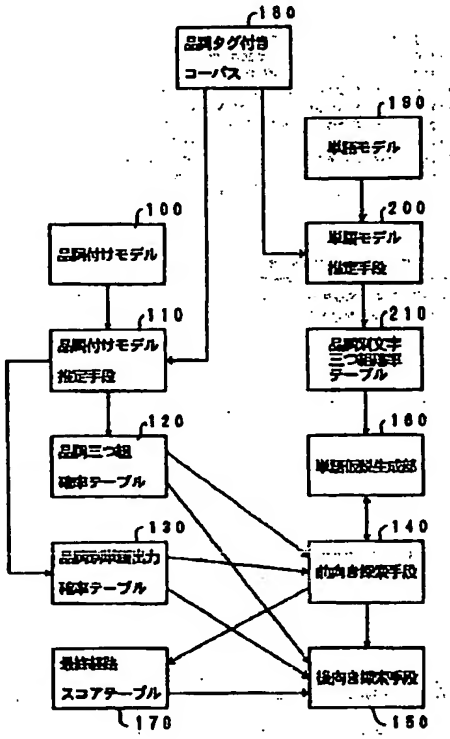
pos:=first(poses);
poses:=rest(poses);
ステップ２０７）オープンカテゴリリストの終わりに達したかどうかを調べる。もしそうであれば、ステップ２０３に移行する。そうでなければステップ２０８以降の処理を行う。
【０１２５】pos==nil ？
ステップ２０８）まず、品詞別の文字三つ組確率を用いて、表記が“form”で品詞が“pos ”である単語の出現確率を求め、変数“prob”に代入する。関数“wordprob-with-word-model”は、表記と品詞を引数とし、品詞別文字三つ組確率から構成される単語モデルを用いて求めた品詞別単語出力確率を返す。次に、表記と品詞と確率から構成されるリストを変数“word-hypo ”に代入し、これを単語仮説リスト“word-hypos”の先頭に加え、ステップ２０７に移行する。
prob:=wordprob-with-word-model(from,pos);
word-hypo:=list(form,pos,prob);
push(word-hypo,word-hypos);
ステップ２０９）単語仮説リスト“word-hypos”を品詞単語出力確率の大きい順にソートする。
【０１２６】sort-by-prob(word-hypos);
図１１は、引数として与えられた文字列の左端から始まる部分列に対して単語仮説を生成した例を示す。
【０１２７】同図において、入力文に未知語がある場合でも出力は、上記の図９のような形式になる。図１１の例では『転送』という単語がシステムには登録されていないが、単語モデルに基づいて生成れた単語仮説の中には、『転送』が「サ変名詞」であるような正解も含まれている。最終的な形態素解析結果は本発明の前向き探索、後向き探索を用いて未知語の前後の部分との関係を考慮した上で決定するため、単語仮説生成の段階で正解が一候補である必要はない。
【０１２８】単語が全て辞書に登録されている場合に最尤候補（最も確率が大きい形態素解析候補）を求めるだけであれば、後向きA*探索は必要ないが、後向きA*探索（グラフ探索）を用いることにより、辞書に未登録の単語も形態素解析の対象とすることが可能となる。
【０１２９】なお、上記の実施例では、統計的言語モデルとしてトライポス（tri-POS)モデルを使用したが、この例に限定されることなく、トライクラス(triclass)、トライ- タグ(tri-tag)、トライ・グラム(tri-Ggram)等を用いてもよい。
【０１３０】なお、本発明は、上記実施例に限定されることなく、特許請求の範囲内で種々変更及び応用が可能である。
【０１３１】
【発明の効果】上述のように、本発明によれば、品詞タグ付きコーパスから推定した品詞三つ組確率と品詞別単語出力確率から構成される品詞付きモデル、品詞タグ付コーパスから推定した品詞別文字三つ組確率から構成される単語モデル、単語モデルに基づく単語仮説生成、動的計画法を用いた前向き探索、及びA*アルゴリズムを用いた後向き探索により品詞列と単語列の同時確率を最大化する入力文の形態素解析候補、即ち、単語分割と品詞付与の組を入力文が未知語を含む場合でも最も尤もらしい候補から順に任意の数だけ出力できるので、高い精度を持ち、頑強でかつ柔軟なインタフェースを持つ日本語形態素解析装置を実現できる。
【図面の簡単な説明】
【図１】本発明の原理構成図である。
【図２】本発明の原理を説明するためのフローチャートである。
【図３】本発明の一実施例の日本語形態素解析装置のモジュール図である。
【図４】タグ付コーパスの例を示す図である。
【図５】品詞三つ組確率の例を示す図である。
【図６】品詞別単語出力確率を示す図である。
【図７】本発明の一実施例のＮ－ｂｅｓｔアルゴリズムのためのデータ構造を示す図である。
【図８】本発明の一実施例の前向きＤＰ探索を示すフローチャートである。
【図９】本発明の一実施例の後向きA*探索を示すフローチャートである。
【図１０】本発明の一実施例の形態素解析候補の例を示す図である。
【図１１】本発明の一実施例の単語仮説を生成する動作のフローチャートである。
【図１２】本発明の一実施例の単語仮説の例を示す図である。
【符号の説明】
１　前向きＤＰ探索部
２　後向きA*探索部
３　品詞タグ付コーパス
４　品詞付けモデル推定部
５　品詞三つ組確率テーブル
６　品詞別単語出力確率テーブル
７　単語モデル推定部
８　品詞別文字三つ組確率テーブル
９　単語仮説生成部
１０　最適経路スコアテーブル
１００　品詞付けモデル
１１０　品詞付モデル推定手段
１２０　品詞三つ組確率テーブル
１３０　品詞別単語出力確率テーブル
１４０　前向き探索部
１５０　後向き探索部
１７０　最適経路スコアテーブル
１８０　品詞タグ付きコーパス
１９０　単語モデル

２００　単語モデル推定手段

２１０　品詞別文字三つ組確率テーブル

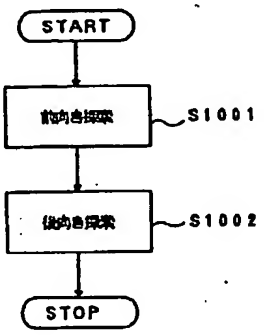
【図１】

本発明の原理構成図



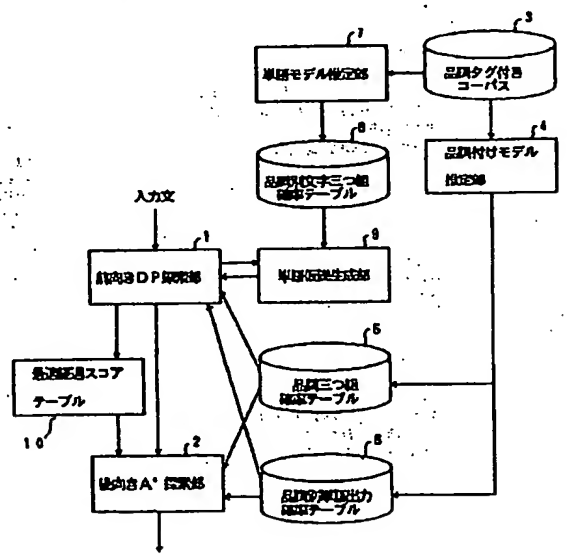
【図２】

本発明の原理を説明するためのフローチャート



【図３】

本発明の一実施例の日本語形態素解析装置のモジュール



【図６】

品詞別単語出力確率を示す図

```
・・・

(サ変名詞　検挙　0.00041017227235433886773)
(サ変名詞　安心　0.00697282863002461037191)
(サ変名詞　案内　0.00615258406531583252681)
(サ変名詞　位置　0.00082094454470877773546)
(サ変名詞　依頼　0.00041017227235433886773)
(サ変名詞　意見交換　0.00041017227235433886773)
(サ変名詞　移動　0.00205088135177194417681)
(サ変名詞　一括　0.00123051681708316854911)
(サ変名詞　一括　0.00041017227235433886773)
(サ変名詞　一致　0.00123051681708316854911)

・・・

(格助詞　から　0.34401697150392765672)
(格助詞　が　0.09966024941293043387471)
(格助詞　って　0.00200876668710624370781)
(格助詞　で　0.09311392665372971983811)
(格助詞　と　0.11203466003844579652)

・・・
```

【図４】

品詞タグ付きコーパスの例を示す図

```
3001-100
もしもし/感動詞 、/記号 通訳電話国際会議事務局/固有名詞 です/助動詞・終止
  か/終助詞 ?/記号

3001-200
はい/感動詞 、/記号 そう/副詞 です/助動詞・終止 。/記号

3001-300
会議/普通名詞 に/格助詞 申込み/本動詞・連用・五段 たい/助動詞・連体
  の/準体助詞 です/助動詞・終止 か/接続助詞 。/記号

3001-400
はい/感動詞 、/記号 登録/普通名詞 用紙/普通名詞 は/係助詞 既に/副詞
  お/接頭語 持ち/本動詞・連用・五段 でしょ/助動詞・未然 う/助動詞・終止
  か/終助詞 ?/記号

3001-500
いいえ/感動詞 、/記号 まだ/副詞 です/助動詞・終止 。/記号
```

【図５】

品詞三つ組確率を示す図

```
・・・

(サ変名詞 補助動詞・終止・サ変 格助詞 0.3142857142857142 7937)
(サ変名詞 補助動詞・終止・サ変 記号 0.20000000000000000111)
(サ変名詞 補助動詞・終止・サ変 終助詞 0.22857142857142855429)
(サ変名詞 補助動詞・終止・サ変 助動詞・連体 0.08571428571428571 5079)
(サ変名詞 補助動詞・終止・サ変 接続助詞 0.05714285714285714 1073)
(サ変名詞 補助動詞・終止・サ変 副助詞 0.05714285714285714 1073)
(サ変名詞 補助動詞・終止・サ変 並立助詞 0.02857142857142857 0536)
(サ変名詞 補助動詞・終止・サ変 補助動詞・終止・サ変 0.02857142857142857 0536)
(サ変名詞 補助動詞・終止・下一 格助詞 0.83333333333333332966)
(サ変名詞 補助動詞・終止・下一 記号 0.16666666666666665741)

・・・

(格助詞 普通名詞 サ変名詞 0.00237863059304818561624)
(格助詞 普通名詞 格助詞 0.55819948443386868875)
(格助詞 普通名詞 感動詞 0.00019829486887346800366)
(格助詞 普通名詞 記号 0.02458853866683125118995)
(格助詞 普通名詞 係助詞 0.09617291294864167 8388)

・・・
```

【図７】

本発明の一実施例のＮ－ｂｅｓｔアルゴリズムのための

データ構造を示す図

ｐａｒｓｅ構造	
ｓｔａｒｔ ｅｎｄ ｐｏｓ ｎｔｈ－ｏｒｄｅｒ－ｓｔａｔｅ ｐｒｏｂ－ｓｏ－ｆａｒ	入力文字列における単語の開始位置 入力文字列における単語の終了位置 単語の品詞 最後の二つの単語の品詞のリスト 文頭からの最適部分解析スコア

ｗｏｒｄ構造	
ｆｏｒｍ ｐｏｓ ｐｒｏｂ	単語の表記 単語の品詞 品詞別の単語の出力確率

ｐａｔｈ構造	
ｐａｒｓｅ ｐｒｅｖｉｏｕｓ ｃｏｓｔ－ｓｏ－ｆａｒ ｔｏｔａｌ－ｃｏｓｔ	ｐａｒｓｅ構造 直前のｐａｔｈ構造へのポインタ 初期状態からのコスト 初期状態から最終状態までのコスト

【図１０】

本発明の一実施例の形態素解析結果の例を示す図

```
> (morph-n-best "会議に申し込みたいのですか。")
-25.62409083687518
会議/普通名詞 に/格助詞 申し込み/本動詞・連用・五段 たい/助動詞・連体
  の/準体助詞 です/助動詞・終止 か/接続助詞 。/記号
-27.28353282681325
会議/普通名詞 に/格助詞 申し込み/普通名詞 たい/助動詞・連体 の/準体助詞
  です/助動詞・終止 か/接続助詞 。/記号
-31.42822072581427
会議/固有名詞 に/格助詞 申し込み/本動詞・連用・五段 たい/助動詞・連体
  の/準体助詞 です/助動詞・終止 か/接続助詞 。/記号
```

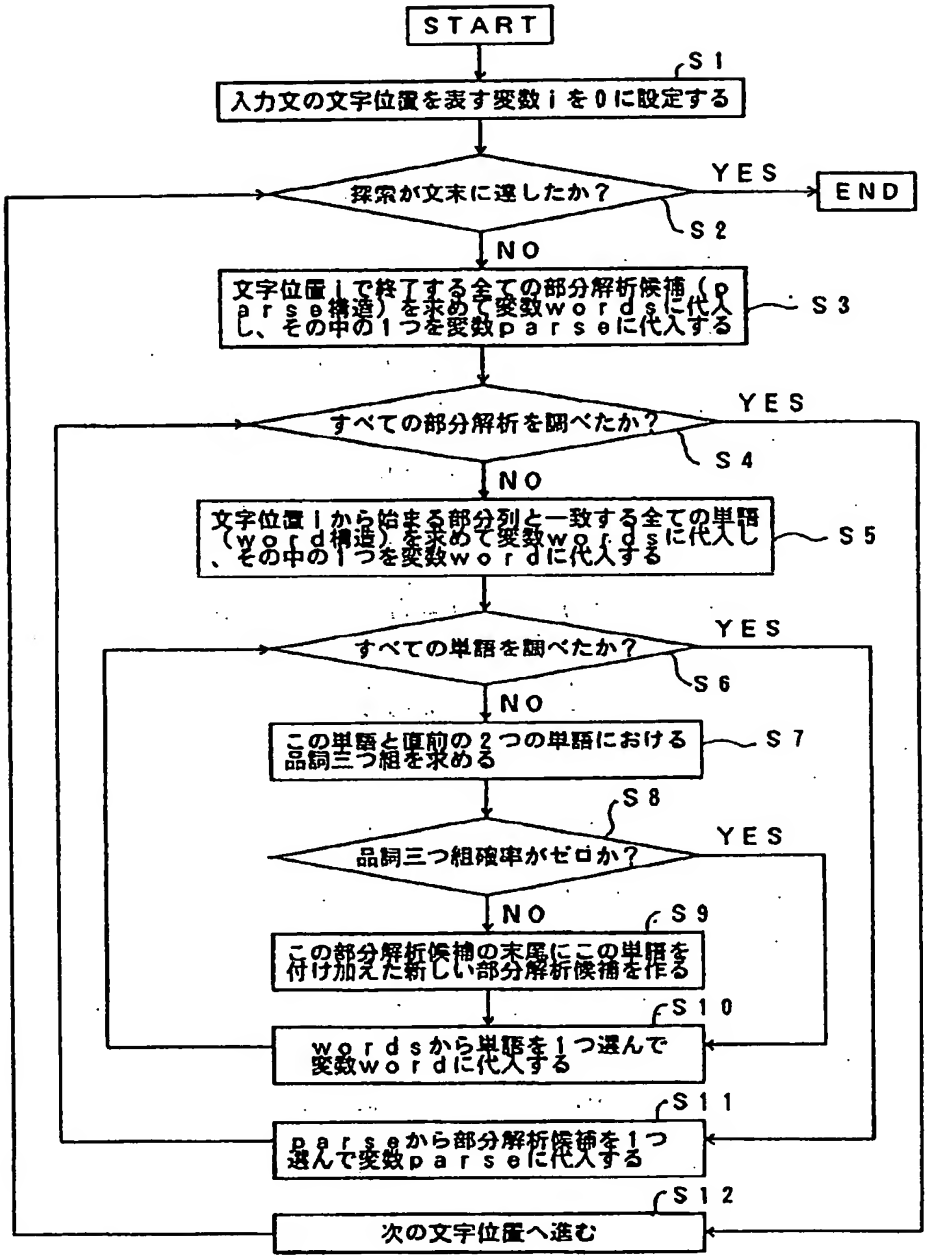
【図１２】

本発明の一実施例の単語候補の例を示す図

```
> (get-leftmost-substrings-with-word-model "を転送してもらって下さい。")
((を サ変名詞 2.6194576873598691E-7)
 (を 固有名詞 2.3449215070189367E-8)
 (を 普通名詞 7.0243990747133745E-9)
 (を 数詞 2.3766509750985687E-9)
 (を転 サ変名詞 6.7281659813601 67E-12)
 (を転 固有名詞 8.9288423481071 83E-14)
 (を転送 サ変名詞 1.6240106526171430E-14)
 (を転 普通名詞 1.1297031867281986E-14)
 (を転 数詞 6.8834934961320 7E-18)
 (を転送し サ変名詞 1.9406358811623394E-18))
> (get-leftmost-substrings-with-word-model "転送してもらって下さい。")
((転 サ変名詞 2.6194576873598691E-7)
 (転 固有名詞 2.3449215070189367E-8)
 (転 普通名詞 7.0243990747133745E-9)
 (転 数詞 2.8766509750985687E-9)
 (転送 サ変名詞 5.7068749302614 15E-10)
 (転送 普通名詞 4.7358228004878569E-13)
 (転送 固有名詞 8.9288423481071 84E-14)
 (転送し サ変名詞 7.2888133442654 52E-14)
 (転送 数詞 6.6854934961320 7E-18)
 (転送し 普通名詞) 2.4530233052613518E-17)
```

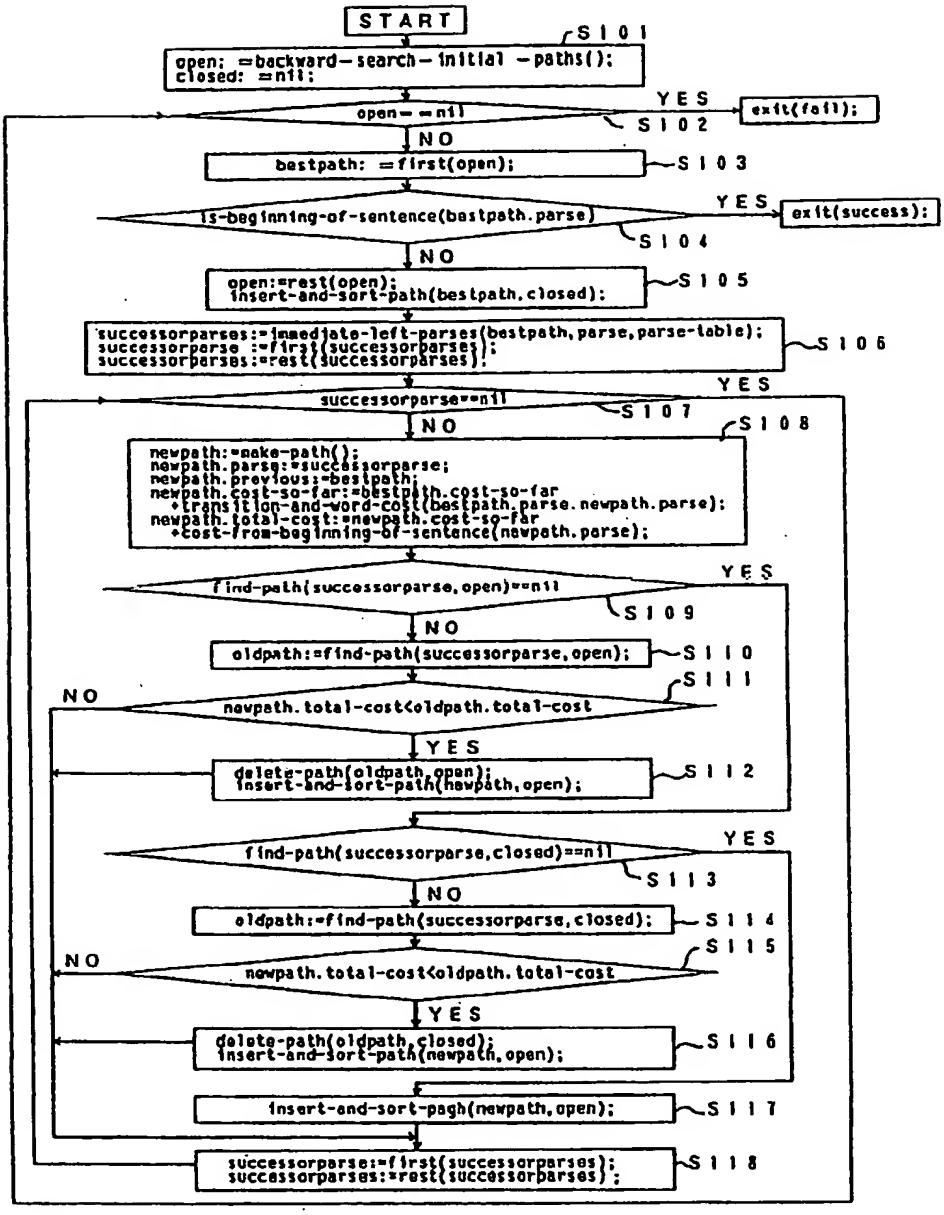
【図８】

本発明の一実施例の前向きＤＰ探索を示すフローチャート



【図９】

本発明の一実施例の後向きA* 探索を示すフローチャート



【図１１】

本発明の単語仮説を生成する動作のフローチャート